SQL skills used in this project to aggregate the data as required. Specific used functions are-EXTRACT(MONTH FROM order_date) for month. GROUP BY, SUM() for revenue, COUNT(DISTINCT order_id) for volume, ORDER BY for sorting, and then Limit results for specific time periods.

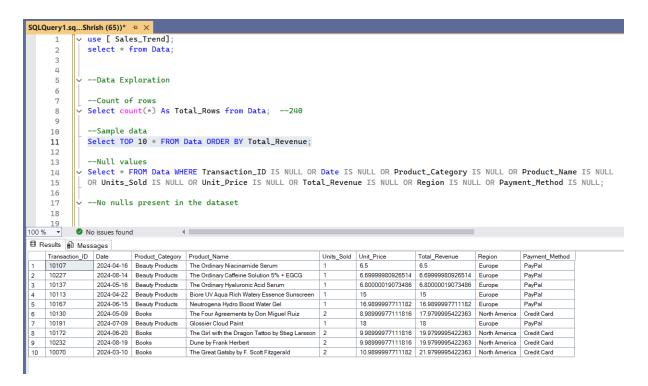
Queries Output-

--Count of rows
Select count(*) As Total Rows from Data;

```
SQLQuery1.sq...Shrish (65))* 垣 🗙
            use [ Sales Trend]:
            select * from Data
     4
            --Data Exploration
     5
     7
            --Count of rows
            Select count(*) As Total_Rows from Data; --240
            --Sample data
    10
    11
            Select TOP 10 * FROM Data ORDER BY Total_Revenue;
    12
    13
           / Select * FROM Data WHERE Transaction_ID IS NULL OR Date IS NULL OR Product_Category IS NULL OR Product_Name IS NULL
    14
           OR Units_Sold IS NULL OR Unit_Price IS NULL OR Total_Revenue IS NULL OR Region IS NULL OR Payment_Method IS NULL;
    15
    16
           --No nulls present in the dataset
    18

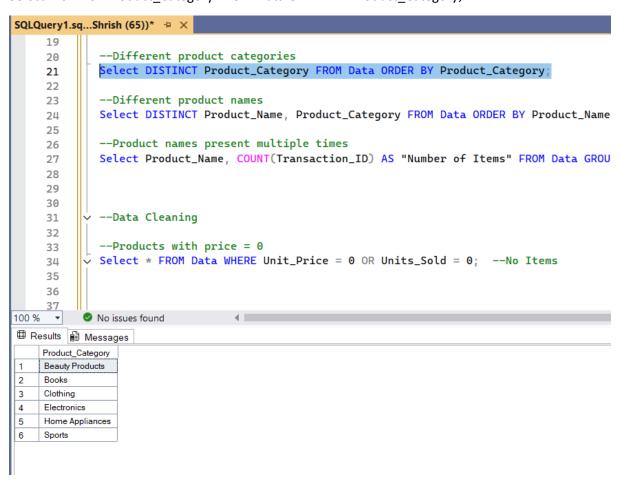
    ⊞ Results  
    ■ Messages
   Total_Rows
   240
```

Select TOP 10 * FROM Data ORDER BY Total_Revenue;

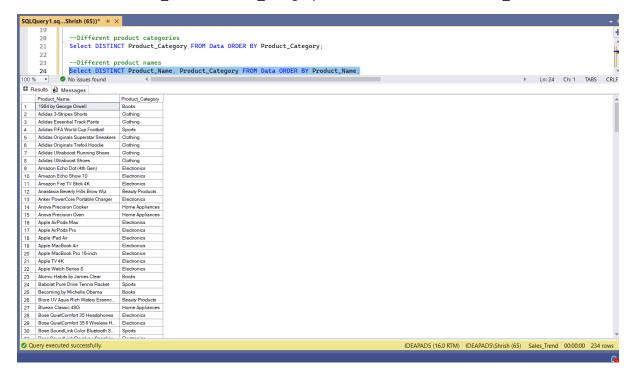


Select * FROM Data WHERE Transaction_ID IS NULL OR Date IS NULL OR Product_Category IS NULL OR Product_Name IS NULL OR Units_Sold IS NULL OR Unit_Price IS NULL OR Total_Revenue IS NULL OR Region IS NULL OR Payment_Method IS NULL;

Select DISTINCT Product_Category FROM Data ORDER BY Product_Category;

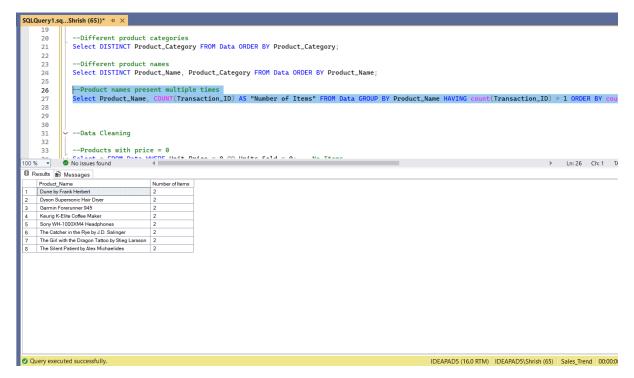


Select DISTINCT Product_Name, Product_Category FROM Data ORDER BY Product_Name;

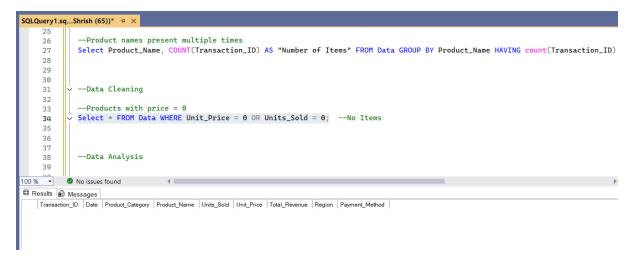


--Product names present multiple times

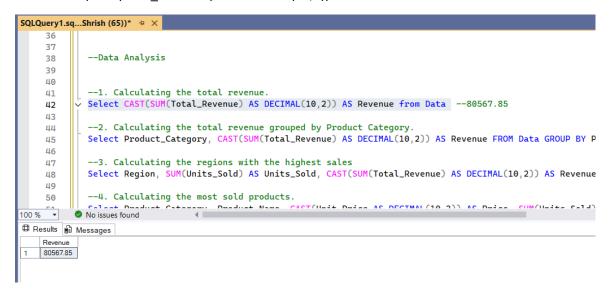
Select Product_Name, COUNT(Transaction_ID) AS "Number of Items" FROM Data GROUP BY Product_Name HAVING count(Transaction_ID) > 1 ORDER BY count(Transaction_ID) DESC;



Select * FROM Data WHERE Unit_Price = 0 OR Units_Sold = 0;



Select CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data



--2. Calculating the total revenue grouped by Product Category.

Select Product_Category, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue FROM Data GROUP BY Product_Category, Total_Revenue ORDER BY Total_Revenue Desc;



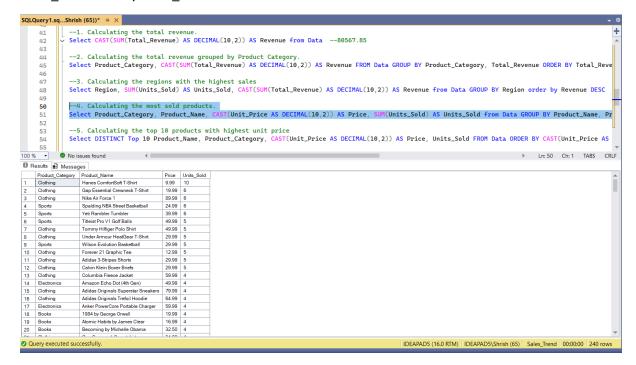
--3. Calculating the regions with the highest sales

Select Region, SUM(Units_Sold) AS Units_Sold, CAST(SUM(Total_Revenue) AS DECIMAL(10,2)) AS Revenue from Data GROUP BY Region order by Revenue DESC



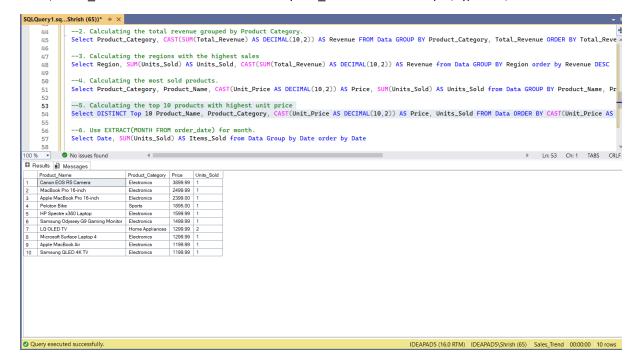
--4. Calculating the most sold products.

Select Product_Category, Product_Name, CAST(Unit_Price AS DECIMAL(10,2)) AS Price, SUM(Units_Sold) AS Units_Sold from Data GROUP BY Product_Name, Product_Category, Unit_Price, Units_Sold order by Units_Sold DESC



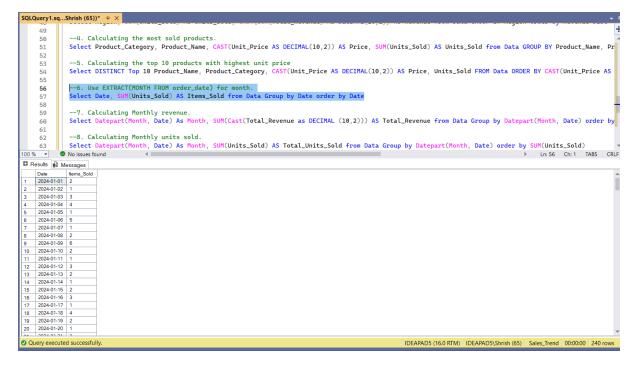
--5. Calculating the top 10 products with highest unit price

Select DISTINCT Top 10 Product_Name, Product_Category, CAST(Unit_Price AS DECIMAL(10,2)) AS Price, Units Sold FROM Data ORDER BY CAST(Unit_Price AS DECIMAL(10,2)) DESC;



--6. Use EXTRACT(MONTH FROM order_date) for month.

Select Date, SUM(Units_Sold) AS Items_Sold from Data Group by Date order by Date



--7. Calculating Monthly revenue.

Select Datepart(Month, Date) As Month, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(Month, Date) order by Total_Revenue DESC



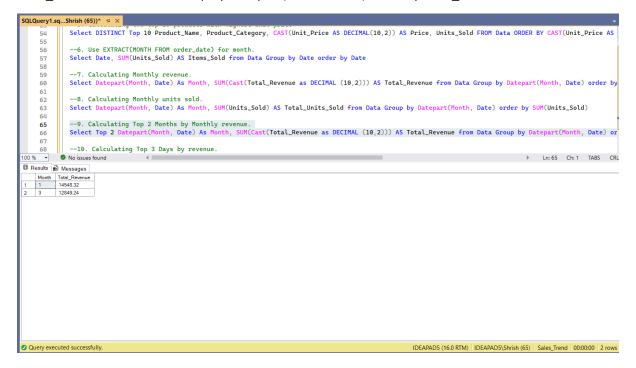
--8. Calculating Monthly units sold.

Select Datepart(Month, Date) As Month, SUM(Units_Sold) AS Total_Units_Sold from Data Group by Datepart(Month, Date) order by SUM(Units_Sold)



--9. Calculating Top 2 Months by Monthly revenue.

Select Top 2 Datepart(Month, Date) As Month, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(Month, Date) order by Total_Revenue DESC



--10. Calculating Top 3 Days by revenue.

Select Top 3 Datepart(WEEKDAY, Date) As WEEKDAY, SUM(Cast(Total_Revenue as DECIMAL (10,2))) AS Total_Revenue from Data Group by Datepart(WEEKDAY, Date) order by Total_Revenue DESC



--11. Calculating the Weekdays having highest units sold.

Select Datepart(Weekday, Date) As Weekday, SUM(Units_Sold) AS Total_Units_Sold from Data Group by Datepart(Weekday, Date) order by SUM(Units_Sold)

